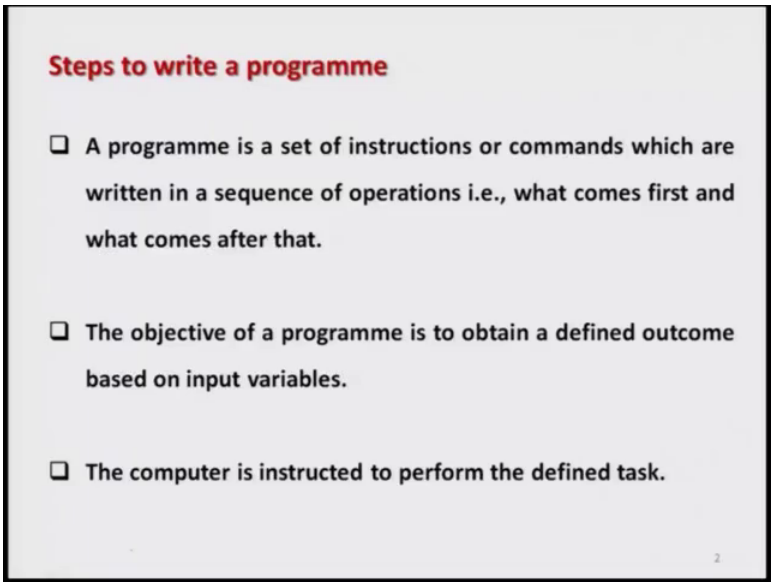
Introduction to R Software
Prof. Shalabh
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

Lecture - 41
Introduction to Programming with Examples

Welcome to the next lecture on Introduction to R Software course. Now you can recall that in all the earlier lectures, we have learnt many things some basic fundamental their syntax their executions. And we have learnt that: what are the different components which can be used to do a specified job. Now in this lecture and in the next lecture we are going to use all those commands and we would like to write a program.

So, in this lecture I am going to first explain you what is program and what are the things that we really try to do inside a program on when we try to write a program. Then I will take a say several examples to explain you those concept and their implementation right. So, the first question comes what is a program.

(Refer Slide Time: 01:13)



So, if you see a program is simply a set of instructions or commands which are written in a sequence of operations that is what comes first and what comes after that.

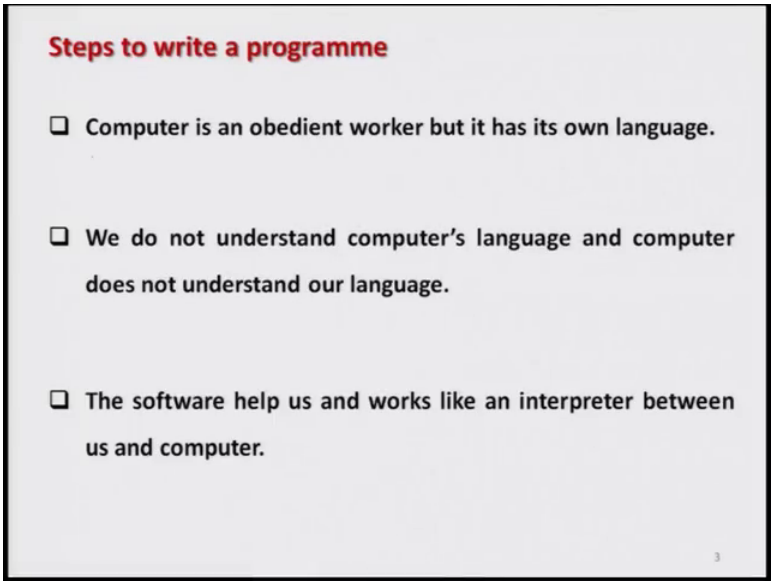
This means what you see whenever we want to do any task. We have to define what we really want as an outcome and in order to obtain that outcome, we have to give some

instructions and those instructions are given in a sequence; that means, whatever the instruction comes first that is given first and whatever are the instruction that will come after that that is given after that. This is what we mean that the instructions are given in a sequence all right. And whenever you are trying to write down a program this program is going to be a compilation of all those instructions which are needed to execute the predefined task right.

So, whenever you are trying to write down the program first of all the objective of a program is defined and that is defined in terms of outcome. And this outcome is based on some input variables, this means whenever we are trying to do something we have to divide the entire program into first 2 parts. What is outcome and what is input right. And based on the choice of output and input we try to instead the computer to do a particular task or a define task. And that is what we called as a program; now when we are trying to instruct the computer that mister computer please do this task for me. So, when we are trying to give the instructions to the computer, then we have to understand one thing the computer is an obedient worker whatever we are instructing it will do the same job and it will try to take the minimum time.

But this all depends on us and the basic problem is this.

(Refer Slide Time: 03:27)

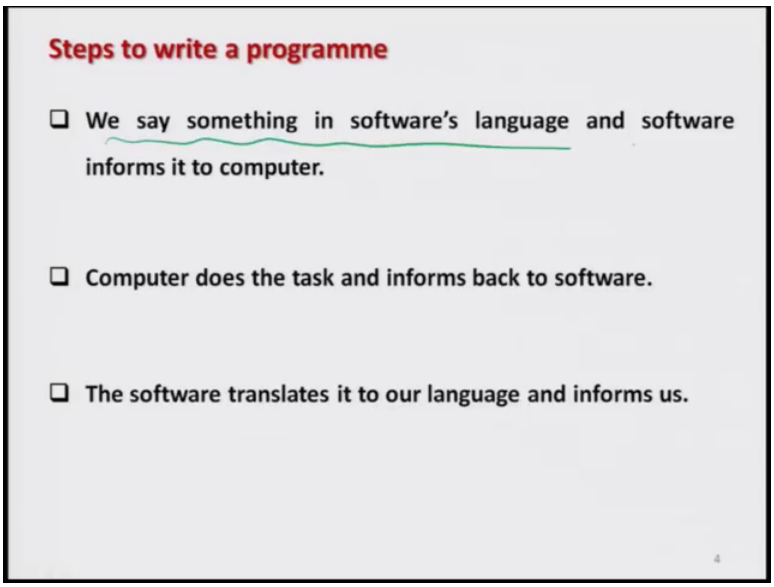


Whenever we are trying to instruct the computer has it is own language and computer does not understand the language like Hindi or English. So, the problem here is that we

are going to take the work from computer, and we do not understand computers language and the computer who is going to take our instruction to do a job the computer does not understand our language. So, the story is now very simple we do not understand computer language computer do not understand our language. So, how to now work? In order to help the software comes into picture. And the software helps us and it works just like an interpreter between us and the computer.

Whatever we are saying the software we try to translate into the computers language and whatever computer is saying it will try to translate to our language. And that is the advantage of using a software or a programming language. So, whenever we are trying to use a software, the software has his own language and which we have to understand. So, whatever we want to get done we say something in software language and then the software in forms those instructions to computer.

(Refer Slide Time: 05:27)



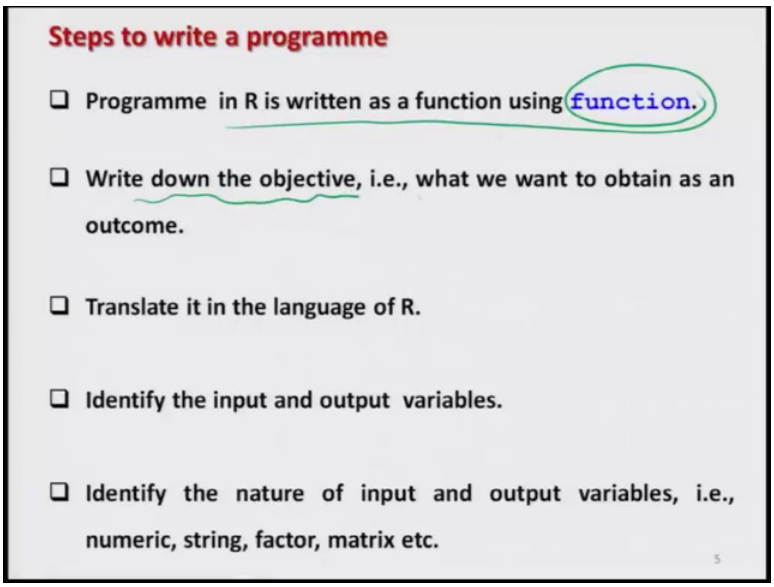
Now for example, if you see what is the software language now you have learnt. So, many instructions which you would like to give to R to do a particular job for example, prend or say format or say function and so on.

So, these are the languages which are software understand. So, now, if we would like to give some instruction using the language for example, R and the software will translate it to the computers language and it will inform the computer that please do this thing. Now computer does the task and then the computer in forms it back to the software. Now

software is once again translating it to our language. So, the software translates it to our language and then it inform us. So, what is really happening, we are trying to say something to the computers through the software language then computer is understanding about instructions then computer is trying to do a job and then the computer is informing it to the software and then software is informing us.

So, that is a very simple philosophy of a program. Now when we are talking about the R software, then we have now understood that R software has it is own language which has a special types of commands they have their own syntax, and there is a way to use them that we have learnt up to now. So, whenever we are trying to write a program in R, then they are written using the command function you may recall that we had once discussed that to be how to write a function. So, whatever we say in some language that we want to write a program in R that is equivalent to saying that we want to write a function and in order to write down the function the first step is that write down the objective.

(Refer Slide Time: 07:39)



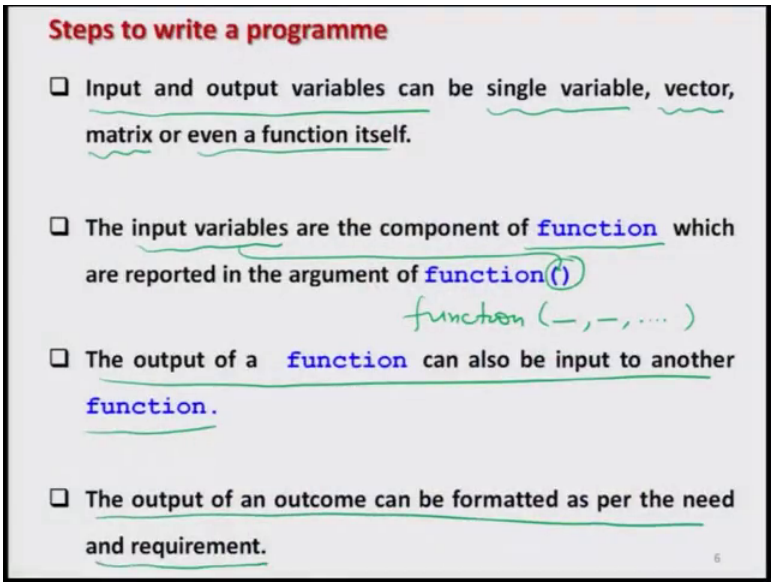
That is what we want to obtain as an outcome of that function and whatever we want to obtain that is translate it in the language of R using the syntax of different commands and that is executed by R.

So, in order to write a program, the first step is that we have to identify the input and output variables. The input variables are given and outcome is obtained. So, whenever we are trying to identify the input and output variables, then we also have to identify the

nature of those variables. For example, those variables can be numeric they can be a string they can be factor they can be matrix or say anything else or even a combination of that. For example, some input variables can be matrix some input variable can be string some input variable maybe some character and so on, that depends on what is the objective and what we want.

Moreover, this input and output variables there can also be a single variable.

(Refer Slide Time: 08:56)



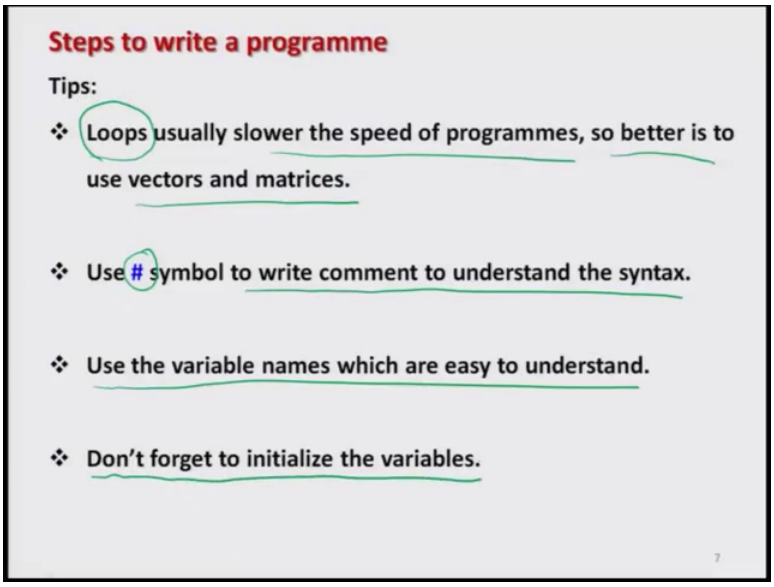
They can be vector valued they can be matrix valued or even they can be a function. And when we want to write down the program, we have to take the help of function. And whenever we are trying to write down the function you remember function has 2 components, one is the syntax f u n c t i o n and then there is an argument; so whatever are the input variables they are written inside the argument. For example, I would try to write down here function inside the argument variable number one separated by comma variable number 2 separated by comma and so on.

And based on that we will try to obtain the outcome one thing you always keep in mind that whatever is the outcome of a function that can also be used as input to another function. And that is the advantage of using here R that is a very important feature of R. So, the output of a function can also be an input to another function. And that is one of the strong feature in this R because whenever we are trying to write down very lengthy program, I can divide the program into smaller components means every component will

be a sort of function and then they will give say some outcome and those outcomes can be combined together in a particular function right. And whenever we are trying to get the outcome we also need that outcome in a particular format.

So, in R we have learnt that the output of an outcome can also be formatted as per the need and requirement you had used different types of command like a spread cat and so on, all those things will come into picture right.

(Refer Slide Time: 10:56)



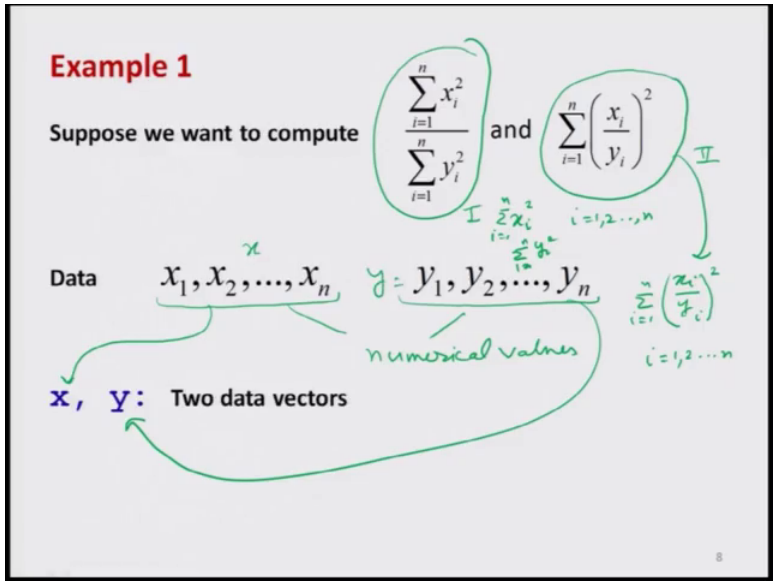
So, now some tips say any programming language in order to repeat a process, we try to use the loop usually the loop slower the speed of programs. So, a better approach is to use the vectors and matrices well, I am not saying that these are the hard and fast rule and you should use only that thing, but try to avoid and one thing you have to also keep in mind that whenever you are trying to write down the program at that point you know each and everything. For example, what this command is doing why I am doing.

But it is possible that after sometime when you try to look into your own program you may not recall that what I was trying to do. So, whenever you are trying to write down a program try to write sufficient number of instructions inside the program so that whenever you are trying to use it again you will recall what you are essentially doing at that time. So, in order to do that thing simply try to use the hash symbol. And try to write comments. So, that you can recall and understand the command mores ever in case if you are trying to send this program to somebody else then he also has to understand it. So, it

is always a better strategy to write down comments to explain the steps in the program. So, that anybody can easily understand it also whenever you are trying to define input and output variables please try to give a name which has some meaning.

For example, in case if I am trying to get a data on say height then give it a name say h e i g h t height do not give it a name, weight well you can give it, but that will create more confusion. So, always try to give a proper name to the variable which can be easily understood by us after sometime or event by others. And whenever you are trying to use a new variable do not forget to initialize them, what do you mean by this we will try to show you in the next program. So, now, we attempt to write a small program for the given objective right.

(Refer Slide Time: 13:16)



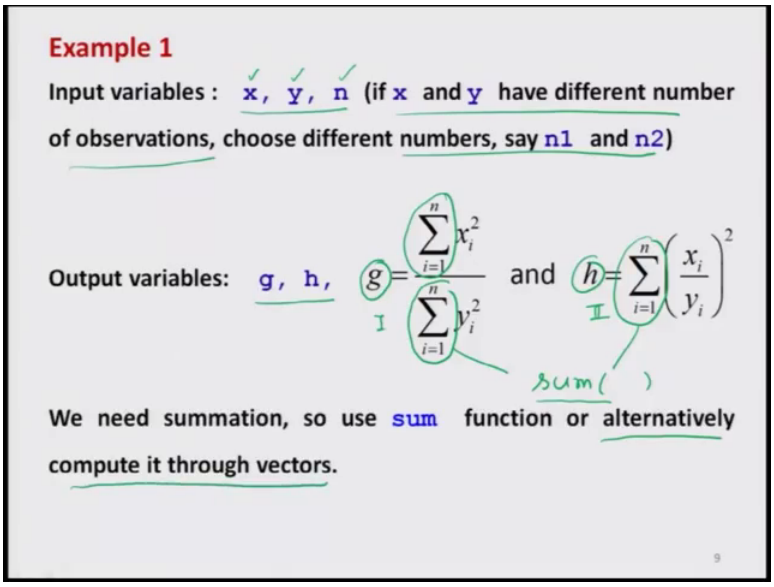
Suppose we want to compute 2 simple functions one is like this summation xi square upon summation yi square. And second function is summation of xi upon yi whole square and in the case number here, first you can see here what you have to do you simply have to take some data.

See here xi you have to square it, and then you have to obtain all the say squares, i goes from one to here n and then you have to make it sum I goes from 1 2 n and the same thing has to be done with summation, i goes from 1 to n yi square. And similarly in this case in the second case what you have to do you have to take the value of xi you have to take the value of yi, then you have to take the ratio and then you have to square it and

this process has to be done for all the values and after this you have to sum all those values together.

So, now means; obviously, we are going to have some numerical values on xi and y is. So, we assume that we have got some data on x say x1 x 2 x n and data on say here y as y1 y 2 y n. So, these are going to be sum numerical values right, and we try to store these numerical values inside the data vectors say here x and say here y. So, that is inside x and this is inside y right. So, x and y are my 2 data vector numerical values. Now we try to identify what is my input variable and what is my output variable. So, here in this case you can see here that there are 3 input variables, x which is the data on x1 x 2 x n y which is the date on y1 y 2 y n, and I also need the number n which is the number of observation present in x and y.

(Refer Slide Time: 15:11)



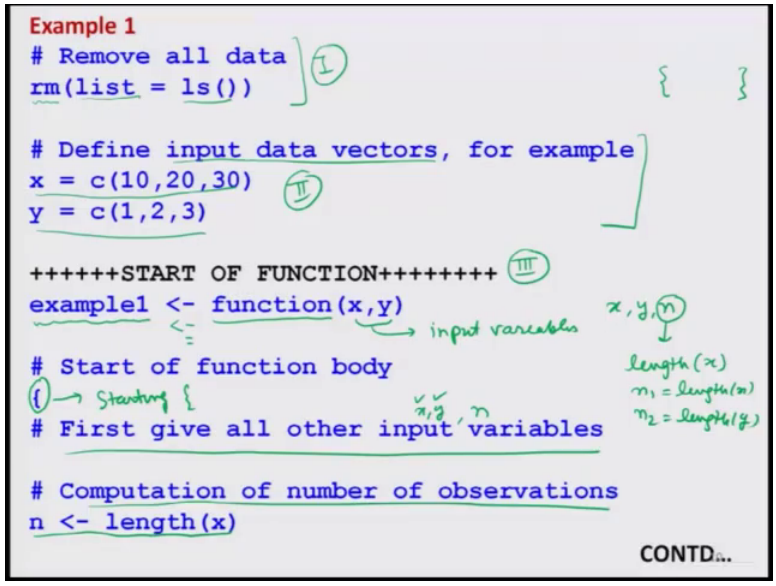
Well in case x and y have got different number of observation then there is no issue we can choose 2 different number say n1 and n 2, but here in our case they are the same right. Now what about the output this output we try to define say by here g and here h g is defining the first function and h is defining the second function. Now, I have here 2 output variables g and h. Mores ever if you try to see here and see here we are actually going to find the sum, we already have done the sum can be found by the function sum and whatever we want to some that has to be given inside the arguments. So, that you

already have done and then whenever we are trying to do here are sum this is summation over xi and y i.

So, there can be a loop also. So, just for the sake of illustration, I am going to use here loop, but alternatively this problem can also be solve using the vectors and matrices. So, both approaches can be done, but I will take here the loop approach. Because my objective is to show you how a program is written now. I come to the program well in this program I am going to continue with several slides, but it does not mean that the program is long the program is very short, but I am trying to write down here as many as possible comments so that you can understand it easily.

After that I will show you that the program is very small. Now let us try to have a look.

(Refer Slide Time: 17:16)



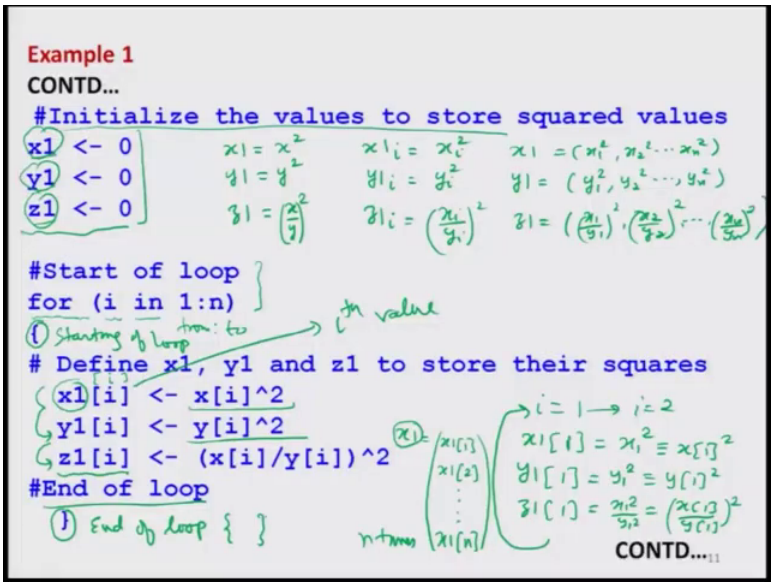
The first step in writing a program is that please try to remove all the data whatever is stored earlier in the program. And in order to do so, you have learnt that the command is R m and inside the argument list is equal to ls and arguments. This will help us for example, if you try to define here say the 2 data vectors x and y and it is possible that earlier you might have used x and y for something else. So, when you are trying to write down the program. So, the computer will not identify that whether you want a new x or you want to use an earlier x. So, better is to clean up everything and then start writing the program.

Then the second step is that we have to define the input variables and then we have to give the input data vectors, for example, here I am trying to give here the values of here x and y just for the sake of illustration right now. After these 2 steps I am in our going to start with writing down the function which is about third step. So, now, first of all I want to give my program a name. So, I tried to give it a name say example one, because we are doing example one as simple as that and then less than and hyphen sign or say (Refer Time: 18:45) sign whatever you want then I have to write down here the function as such then inside the bracket, I have to give all the input variables.

But now if you try to see there is one difference earlier I had told you that there are 3 input variable x y and n, but here I am giving you only 2 y. Because you will see that n is nothing, but this is the number of observations in say x and y and which I can directly compute inside the program using the command length of x or say length of y. And in case if n is different then I can define say n1 is equal to say this here length of x or say n 2 can also be say here length of y right. So, now, after this after writing this thing I start writing the body of the function, and for that I start by writing a curly bracket. Remember one thing that all the instructions which are given inside a function they are written inside the curly bracket.

So, this is my here say starting bracket say starting bracket right. So, the first step is this first try to give all other input variables. So, for example, here I have given say here x and y they are already given, but now and his left that is also an input variable. So, I try to define here n by length of x. So, here I try to compute the number of observation in the data vector x and whatever is the outcome that I am trying to denote by here n right.

(Refer Slide Time: 20:30)



So, we continue further, and now in the next step I try to define here 3 more variables. You can see here x1 y1 and z1. Why they are needed, that will be just clear to you in the next step, but at this moment I am writing that please initialize the values x1 y1 and z1 and why I am giving it the reason is as follows.

That I want to denote x1 to be something like x square so once I write here x1 I this is actually my xi square so I am trying to create here a vector which is going to contain the values of x1 square x 2 square up to here x n square and similarly I am trying to define here another variable here y1 which is something like y square so the ith value in this vector is simply denoting the yi square so when I try to write down the variable y1 this is going to denote a vector containing the value y1 square y 2 square y n square. And then we define another variable here z1, which is trying to contain the values of say x square upon y square something like x upon y whole square.

So, the ith value of z1 is something like xi upon yi whole square so this vector here z1 contains the value x1 upon y1 whole square x 2 upon y 2 whole square and so on up to here say x n upon y n whole square right. And you can see here that these values are needed to compute this here this component, for summation xi square this component for summation y square and this component for summation xi upon yi whole square right. So, now, here in this part I am trying to say I am trying to define here 3 variables x1 y1 and z1 and the initial value of this variable is 0.

So, whenever we are trying to compute the value of x1 y1 and z1, they are going to be replaced by the new value and earlier value is 0. So, that is why I need to initialize these 3 variables. So, now, I start my here loop. Why should I use here loop? Because I want to compute the value of x1 i y1 i and z1 I say small n number of time. So, I use here for loop you may recall that as syntax for the for loop is for inside the argument starting say this variable which is going to control the loop say I in from then colon 2. So, that will start from one and it will take the value up to here and at an increment of one unit at a time that is what we have learnt.

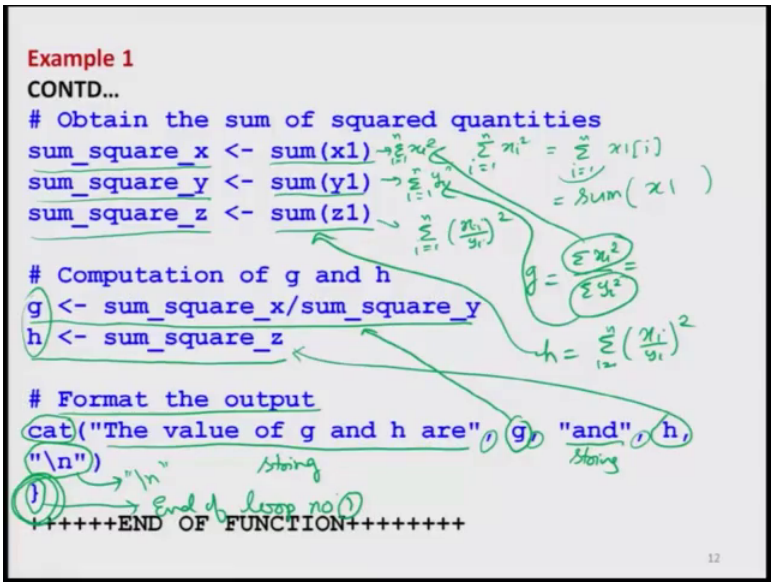
So, now I tried to compute the value of x1 y1 and z1 at ith place. So, far that I try to define here x1 and inside these square brackets, I write down here i. So, that mean this means here ith value. So, I will start from 1, and then it will come to x1 and it will try to compute the first value here something like x1 square. And which I am giving here as say xi whole square right. This is here something like x1 say square. And then the control will come from here to here, and then it will try to compute the first value in the y1 vector say y1 1 y1 square and then it is something like y square bracket one and square, which is here.

Then after computing the second step it will come to the third step. And here it will try to compute the ith value of z1 for example, for I equal to one this z1 will become here z1 1, say x1 square upon y1 square and that is something like say here xi one upon y1 say whole square right and then it will store it here. So, there is now here a for example, a vector here which is taking the first value second value third value and so on. So, here I get the first value.

Now, this control will come back to here and it will try to take the i equal to 2 and this process will be repeated and it will compute the value of x1 y1 and z1. For example, at the second place and so on and this will be repeated n times and then finally, I will have here a value here x1 n right. Now, I have created 3 vector something like x1 y1 and z1 which are trying to store the values of say x1 y1 and z1 at their individual places right and here, I would like to stop the loop. So, one thing you have to notice that the loop has been given inside this to curly brackets, this is the starting of loop and wherever I want to inform my program that please stop the loop here, I say this is end of loop.

So, I try to inform the computer the starting and end point of the loop just by using the say curly brackets like this. So, now, this loop is repeated and then we have got all the values of x1 y1 and z 1.

(Refer Slide Time: 26:52)



Now, what we want we want for example, I goes from 1 to n summation xi square. Now I know this is nothing, but this is i goes from 1 to n then summation xi square is something like x1, i write and for this here summation here we know that there is a function sum. So, here I would like to find out the sum of all the values which are stored in the variable say here x1 right. And that is what I try to do here sum of x1 sum of y1 and sum of z 1.

So, this is going to give a summation xi square I goes from 1 to n the sum of y1 is going to give us a value of i goes from 1 to here n yi square and sum of z1 is going to give us the value i goes from 1 to n xi upon yi whole square. And these values have been stored in say another logical name some underscore say square underscore, x and similarly sum underscore square underscore y and some underscore square underscore z. Now I want to compute my function g and h. So, g if you remember this was summation xi square upon summation yi square.

So, now this is my nothing, but summation xi square has been obtained here and summation yi square has been obtained here. So, I can simply write down here the ratio of the 2 variables which are storing the sum of squares. And similarly for here h this is

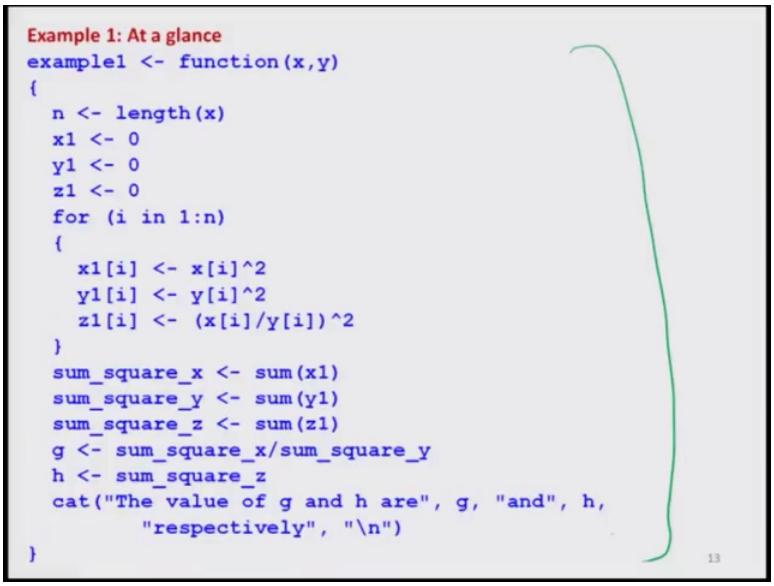
simply here summation i goes from 1 to n xi upon yi whole square and these values have been stored here. So, I try to simply write down here g is equal to this and h is equal to this variable sum underscore square underscore z.

So, now this will compute the value of g and h and we have the outcome for a given value of x and y. Now I need this outcome in a particular format. For example, I want the outcome should be reported like the value of g and h are so this is going to be a string. And then the whatever is the value that is obtained here that should come here then there should be another string say and whatever is the value of here h which is obtained here, that comes over here and then it should change the line. So, now, you see we have learnt that in order to obtain this type of outcome we can use the command cat. Then I will give all those strings and variable separated by this comma.

And they this is a command back slash n, inside a n double coat to change the next line. So, here now we stop and we want to tell our computer now I have done please stop and give us the outcome. For that I write here this another curly bracket which is the end of function. So, if you try to see here I have used here 2 types of curly bracket, let me call it here suppose this is my curly bracket number here one. And then I have used here another curly bracket here say here 2 which is here. So, 2 is ending here this is the end of loop number 2. And now at the end this bracket this is the end of loop number say here 1.

So, whatever loop you are going to start that has to be end right. So, this is our program, but now you may feel that it is a very long program. So, now, I have written this entire program in a single slide.

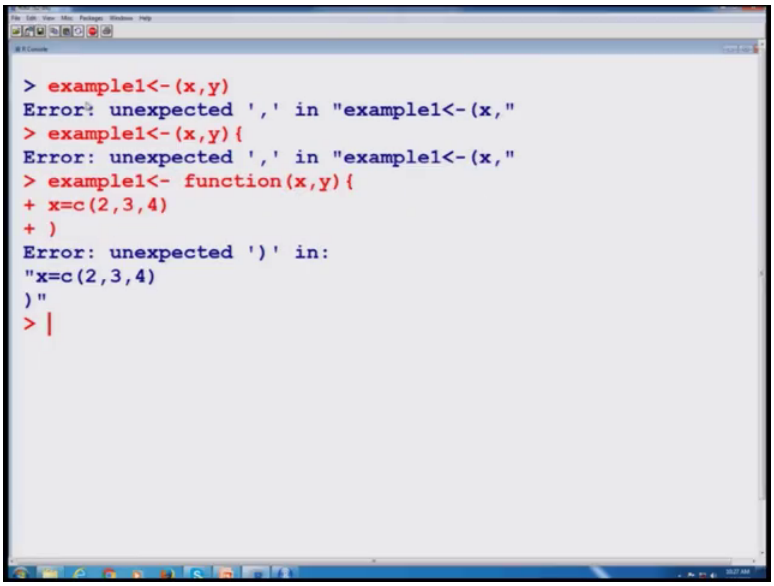
(Refer Slide Time: 31:04)



You can see here this is the same program but here I have removed all the comments and I have also reduced the phone size in little bit, but you can see that is not a very difficult program the only thing is this you have to decide, what are the steps that you want to do and in what order these instructions have to be given so that the computer can understand what is to be done right.

Now, we want to write down this program in the R console in order to do. So, we have 2 options means either I can write down it here in the R console directly, but now you can see here some problems, say for example, if I try to write down here is example one and then I try to give here the input say x and y.

(Refer Slide Time: 31:48)

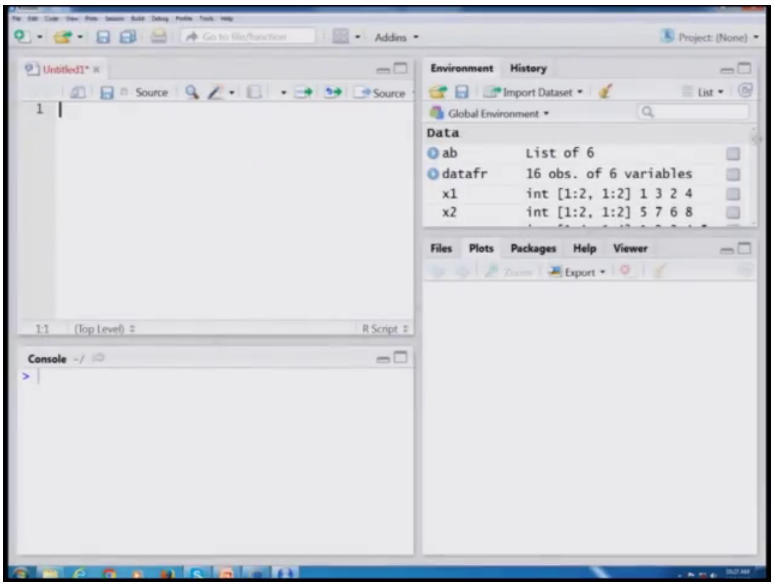


And then I start with q o that, the something wrong right because we have not given here the bracket here. So, as soon as you, but it is something wrong here once again can you identify what is wrong here because you have not given the command here function and then you are not trying to tell this R that the this is going to be a function.

You understood that example one is a function, but computer does not understand. And in order to write down the function you have to give this curly bracket and then you have to start writing. So, as soon as you get here the plus sign; that means, now you have to write down the syntax and instructions. For example, I can write down here x is equal to say here c and say 2 3 and say 4. And now means I have done something wrong suppose I wanted to write here x equal to 20 30 and 40, means I cannot come back here you can see here I cannot come back here. So, now, either I have 2 things. Suppose I try to make here some mistake as soon as I make a mistake this program comes out of the control. And whatever I have typed here that goes into waste.

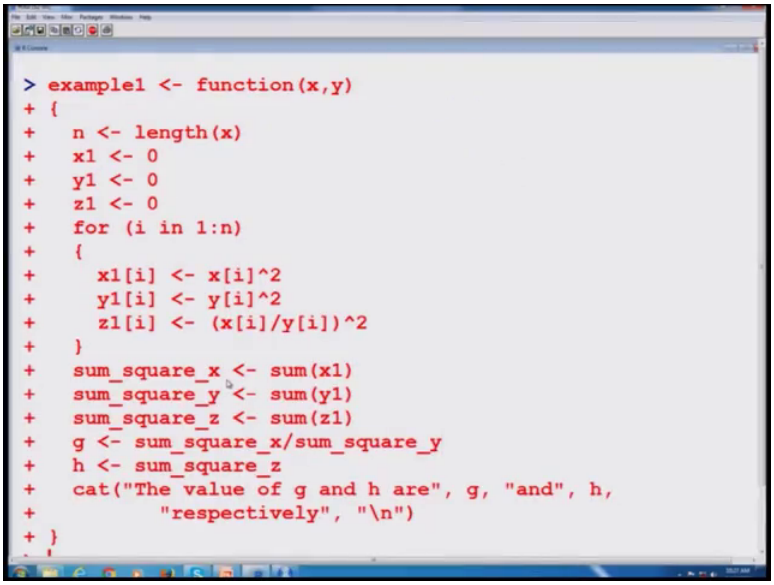
So, it is really not advisable to type the program directly into the R console, you have 2 options that we discussed in the initial part of this course that you can use here the R editor you tried to type the commands over there. Or you try to use the R studio, this will help you more actually right.

(Refer Slide Time: 33:39)



So, I try to do here both the think. So, I have simply copy the entire program and I am simply trying to paste it here.

(Refer Slide Time: 33:49)



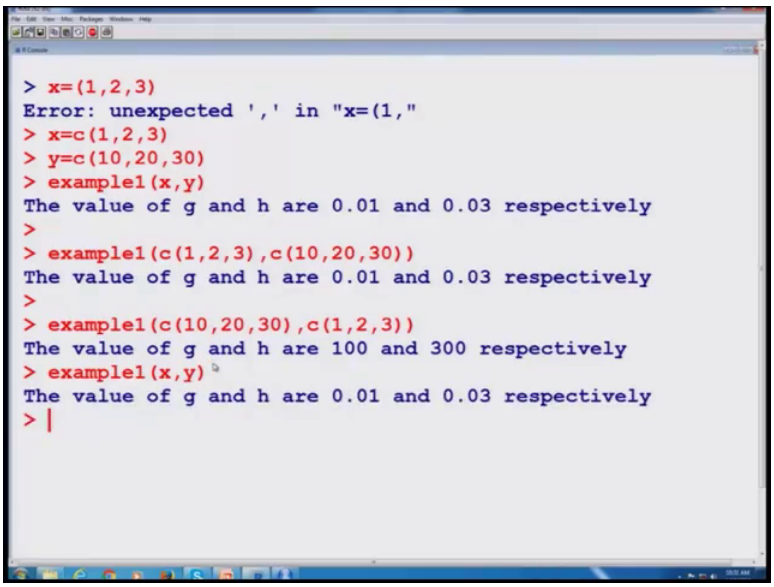
```
> example1 <- function(x,y)
+ {
+   n <- length(x)
+   x1 <- 0
+   y1 <- 0
+   z1 <- 0
+   for (i in 1:n)
+   {
+     x1[i] <- x[i]^2
+     y1[i] <- y[i]^2
+     z1[i] <- (x[i]/y[i])^2
+   }
+   sum_square_x <- sum(x1)
+   sum_square_y <- sum(y1)
+   sum_square_z <- sum(z1)
+   g <- sum_square_x/sum_square_y
+   h <- sum_square_z
+   cat("The value of g and h are", g, "and", h,
+         "respectively", "\n")
+ }
```

So, you can see here whenever you are trying to see this plus sign plus sign means that program is still continuing on the next line. And as soon as you try to say here enter this program is now done right, but if you want to make any changes over here in this R console that is very difficult.

But on the other hand if you try to write down this program in the R studio, you can see here that even if you want to make it here something over here say instead of x square say x cube you can do it. Whatever you if you want to remove anything you can simply remove it right. So, that is the advantage of writing the program inside the R studio the things are more convenient right. And now I want to run the program, but before that I would like to see what is my this program. So, I try to do here something like this suppose I want to see what is my example one program suppose you have type with couple of days earlier and you have forgotten what was there. So, I tried to simply right here example one, and as soon as I enter I get here the entire details that this was my example one function.

Now, I want to use this function. So, far that I have to give here certain values suppose I try to give here say x is equal to suppose the first I make a mistake.

(Refer Slide Time: 35:12)



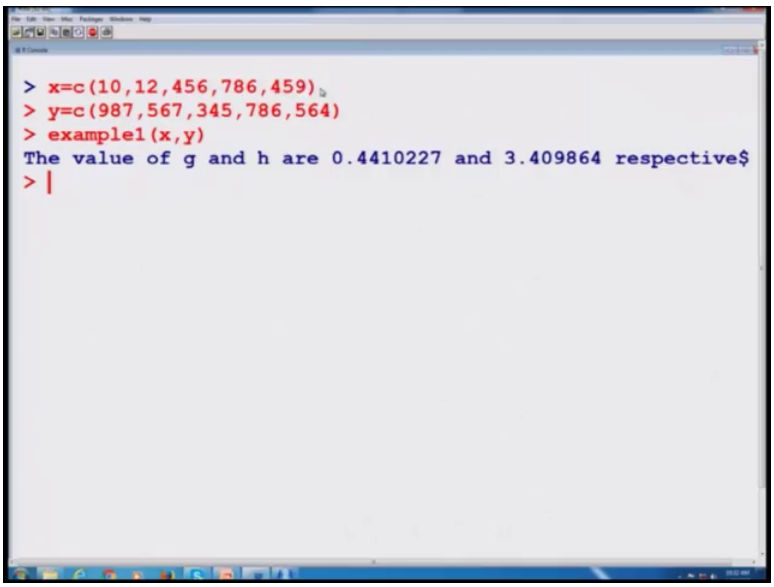
And suppose I try to give here 2 values say x here like this you can see here this is not acceptable you have learn that this has to be with the c command. So, now, x is there and now I tried to write down here the y value also. So, you have to keep in mind that the number of elements in our program in x and y they are the same. So, I have taken here 2 sets of values for x and y, and I want to run the program; that means, I want to find out the value of the h and g using this x and y values.

So, the rule is simply try to write down the name of the function which is example one and try to write down here all the values, whatever have been obtained and you can see here as soon as you enter you get here an outcome this program has already computed the value of g and this part which I have highlighted, this is coming as a string then the value of g has been obtained like this and the value of h has been obtained here like this 0.03 and then it is a string.

Alternative approaches that that inside the example one argument, I can also write the 2 variables one say 2 and here 3 and say another vector here say here c say 10 20 and 30. Now the thing is this; whatever is the order in which you are trying to give the input variable the same order has to be maintained. And you can see here we get the same outcome. On the other hand, if you try to reverse the order for example, here I make here 10 20 and 30 that is the values of y and here, I try to make the values of here x1 2 and 3 you can see here the outcome is changed, because you had defined the function example one in the form of x and y not in the form of y and x.

So, whatever is the first value that is given here that is taken as x and whatever is the second value it is given here in the argument that is taken here as say y right. So, you can see now, here that without doing anything you can obtain different values of here function using the different values of here x and y.

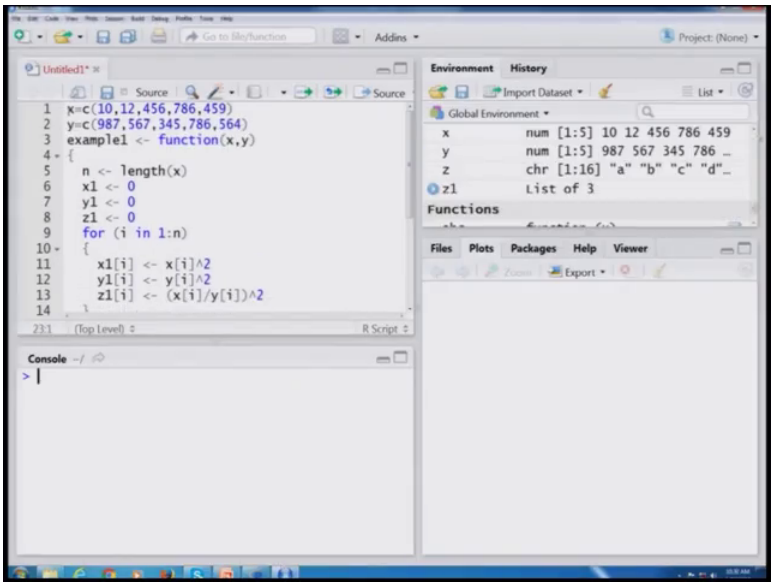
(Refer Slide Time: 37:47)



For example, now suppose I want to change the value of here x suppose, I want to give it here c say 10 12 4 5 6 7 8 6 4 5 9 and so on. And similarly the value of here y here is another vector say 9 8 7 5 6 7 3 4 5 and say 7 8 6 and then say 5 6 4. And you try to use it here and then as soon you say example one it will give you the new values of g and h using these values of x and y, right.

So, this is how we try to write a program and this is how we try to obtain the outcome. The same thing can also be done in the R studio software also right. You simply have to write down here the value of here see here x, for example, I can copy here same thing say here x. And similarly I can copy here the same value here say here y and first I try to means run stored it in the x y try to here run you can see here. Now this is the value of here x.

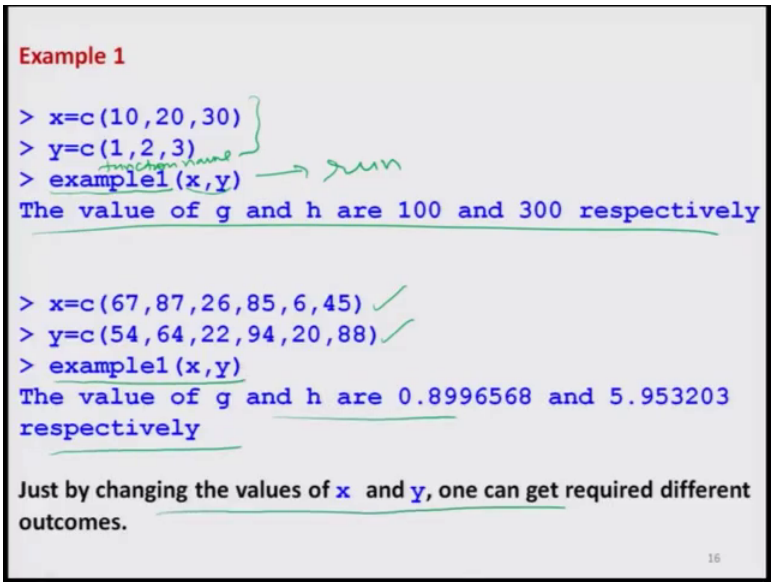
(Refer Slide Time: 39:09)



Now, I tried to give here the value of here y, and suppose I want to run this program. So, I simply have to highlight it here and then I have to write down here.

So, you can see here this is the same outcome that we had obtained on the R console here all right. So, you can see here it is more convenient to work on R studio rather than directly on the R console well you can do it also I am not saying that it is not possible right. Now, we come back to our slides and you can see here, that this is the screenshot of the same thing that we did earlier. And this is another screenshot that when we want to see the details of my function.

(Refer Slide Time: 40:14)



And then we are means I have shown you that we try to take here different values of x and y. And in order to execute or run the program I have to write down the name of the function, function name and then I have to give here all the input variables in the same order. And then here I get this type of what come what we wanted.

Similarly, if I take another set of x and y values and if I try to repeat the same command, I get here different outcome. So, you can see that just by changing the values of here x and y1 can get required different outcomes, that this is the screenshot of the same thing that we did earlier right. So, now, we are done with the first example. And I have try to show you that how the things can really be executed and if you try to observe what I have done, once I had a problem I define my objective that I want to compute g and h.

Then I divided the competition of g and h into different number of steps for example, first I would like to square the values, then I would like to sum the values. And then I have to do this thing for each and every variable that is the logic which I have used and then I simply used a loop to compute all the values of x square y square and x upon y whole square. And then I defined another variable say x1 y1 z1 to store the square values. And after that I simply use the function sum to obtain the sum of the squares and then I wanted the outcome in a particular format. So, I use the command cat.

So, that I can get the outcome in that required format and then we had computed all the values. So, this is the program I have written all the say smaller instructions in a

sequence in a logical sequence, and in the order in which they have to be computed, means if I try to write down first find the square and then I tried to give the input variable the program will not run you can try actually just by changing the order and then (Refer Time: 43:39) means whenever I am trying to write down the program.

Usually, it is not really possible that you can write the hundred percent correct program in the first short. So, there will be some mistake there may be some problems of curly bracket arguments say comma and so on. So, as soon as you run the program the R console will give you some mistakes some error some messages you have to understand those messages and then you have to correct the program and you have to re run it.

So, in the next lecture we will try to take some more examples, till then goodbye.